

---

# Design Document

---

Home Subscriber Server ( HSS )

---

Version 1.0

---

**Objective**

**Document Scope**

The information provided in this document specifies the design details of Home Subscriber Server (HSS). For complete scope of HSS, please see the Project Proposal.

**Table of Contents**

<b>1. References &amp; Abbreviations .....</b>	<b>iv</b>
<b>2. Introduction .....</b>	<b>1</b>
<b>3. HSS Architecture.....</b>	<b>1</b>
<b>4. Components of HSS Architecture.....</b>	<b>3</b>
4.1. Asynchronous Middleware for Protocol Sever (AMPS) .....	3
4.2. Diameter Base Protocol Stack.....	3
4.2.1. Interaction of Diameter Stack with AMPS .....	3
4.2.2. Interaction of Diameter Stack with Main Module .....	4
4.2.3. Interaction of Diameter Stack with Sh and Cx/Dx Modules .....	4
4.3. HSS Main Module .....	4
4.4. Cx/Dx Module and Sh Module.....	4
4.5. Database Module .....	5
<b>5. Typical control and Data Flow in HSS .....</b>	<b>6</b>

# 1. References & Abbreviations

## References

Following is the reference document list, which is related to the information present in this document:

[1] 3GPP TS 29.229 V8.2.0: "Cx and Dx interfaces based on the Diameter protocol; Protocol details".

[2] 3GPP TS 29.228 V8.2.0: "IP Multimedia (IM) Subsystem Cx and Dx interfaces; Signaling flows and message contents".

[3] 3GPP TS 29.329 V8.1.0: "Sh Interface based on the Diameter protocol; Protocol details".

[4] 3GPP TS 29.328 V8.1.0: "IP Multimedia (IM) Subsystem Sh interface; Signaling flows and message contents"

[5] IETF RFC 3588: "Diameter Base Protocol".

## Abbreviations

Following are the abbreviations that have been used in the document:

<b>IMS</b>	IP-Multimedia Subsystem
<b>DBMS</b>	Database Management System
<b>PS</b>	Packet Switched
<b>CS</b>	Circuit Switched
<b>AS</b>	Application Server
<b>CSCF</b>	Call Session Control Function
<b>HSS</b>	Home Subscriber Server
<b>API</b>	Application Programming interface
<b>GPRS</b>	General Packet Radio Service
<b>MSC</b>	Mobile Switching Center
<b>GSM</b>	Global System for Mobile Communication
<b>AMPS</b>	Asynchronous Middleware for Protocol Servers

## 2. Introduction

The Home Subscriber Server, in its basic role, is a centralized database for IMS, Packet Switched (PS) and Circuit Switched (CS) entities. These entities include CSCFs, AS, GPRS Support Nodes and MSCs. The Home Subscriber Server (HSS) is core network entity that plays a very central role in user authentication, authorization and session management. The HSS along with call state control functions completes IMS Control Layer functionality for subscription and session management.

The HSS supports different interfaces of IMS network, PS/GPRS network, CS/GSM network and general purpose IP network. The HSS Server contains user information necessary to support call control mechanism in IMS and other networks (GPRS, GSM or Internet). The Call State Control Functions query HSS over Cx/Dx DIAMETER Interface in IMS network and retrieves subscription data, authenticates a call, authorizes and verifies services profiles related information. This data helps these entities in providing the secure and authorized call control over IMS network. The Application Servers query HSS over Sh Interface for service specific data for a particular subscriber.

In this project we will be implementing the HSS for IMS entities only, and that is a constraint and limit on the current design.

## 3. HSS Architecture

The Home Subscriber Server (HSS) is written using the AMPS framework, which is an Asynchronous Event Based framework that provides following advantages

- Hides the implementation details of Operating System specific API, thus making the code written in AMPS to be portable across various operating systems.
- Provides extensive support for asynchronous and event driven programming paradigm specifically tailored for writing protocol servers for networking and telecommunication domains.

AMPS framework is implemented as a shared library, and lies at the bottom of the architecture. The diameter base protocol stack has been implemented as a shared library, and lies on top of AMPS. The communication between Diameter Base Protocol stack and AMPS is through direct function calls and through callback functions.

HSS itself is implemented as an AMPS module, which we can call the application's main module. The Sh and Cx/Dx interfaces which are part of HSS functionality are implemented as standalone AMPS modules. The functionality that accesses DBMS, is implemented as standalone AMPS module, and is known as Database Module. Any access to DBMS required by any constituent component of HSS will be serviced through the Database module and never through direct DBMS interaction. The following diagram represents the Home Subscriber Server Block Level Architecture.

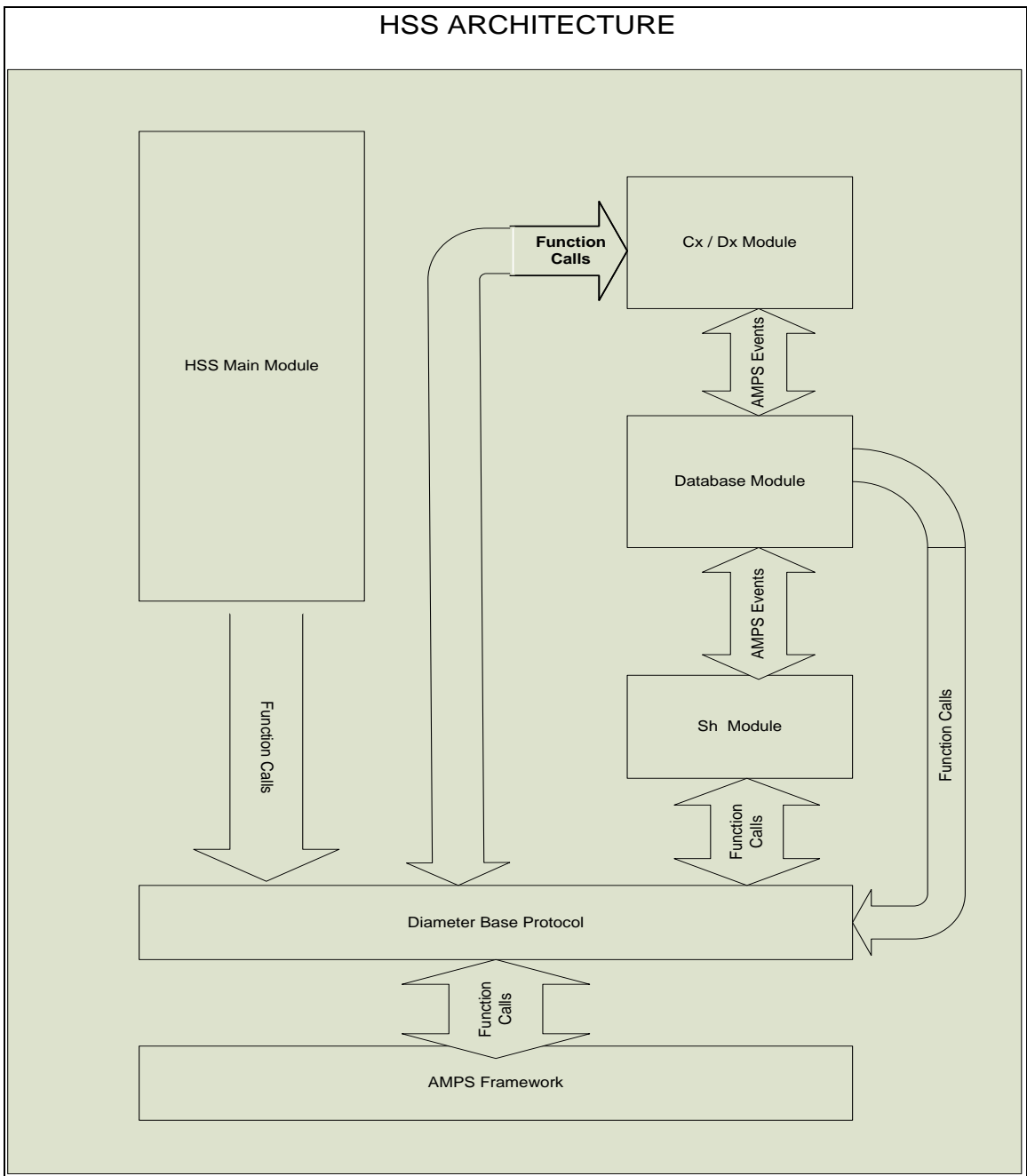


Figure 1 – HSS Architecture

### 4. Components of HSS Architecture

The component of the HSS architecture, as illustrated above, are discussed in detail in the following section.

#### 4.1. Asynchronous Middleware for Protocol Server (AMPS)

Asynchronous Middleware for Protocol Servers (AMPS) is an open source server development platform supports event driven, asynchronous programming paradigm. It is ideally suited for development of high performance application protocol servers in telecommunications and networking domains.

AMPS, with its asynchronous, event-driven programming model and fully abstracted OS dependent layer provides the end-user with following features:

- High performance.
- Supports multi – operating system environment.
- Requires substantially lesser time to develop and market.

AMPS is a powerful middleware to implement protocol servers. Due to above mentioned features, AMPS is selected as the underlying middleware framework to be used in the design and development of HSS. AMPS exist in the form of a shared library to be linked with the main application that is to be developed.

#### 4.2. Diameter Base Protocol Stack

The Diameter Base Protocol Stack is a complete implementation of IETF RFC 3588 specifications, utilizing the AMPS framework. This exists in the form of a shared library that needs to be loaded and configured by its client. The HSS will act as a client of the Diameter Base Protocol Stack. The interaction of Diameter Stack with other components of HSS is further explained below item wise:

##### 4.2.1. Interaction of Diameter Stack with AMPS

The Diameter Stack is a client of AMPS, and makes a direct function call into the shared library of AMPS. Various functionalities i.e. notification of sockets data reception, Diameter Stack registers its callback functions with AMPS. These callback functions are in turn called by AMPS when relevant events or triggers are fired. So, the interaction of Diameter Stack and AMPS is that of mutual direct function calls into each others' shared libraries.

### 4.2.2. Interaction of Diameter Stack with Main Module

The main module of HSS is an AMPS module by nature. It happens to be a client of Diameter Stack. The main module is responsible for the configuration of Diameter Stack shared library. This configuration happens through direct function calls into the Diameter Stack Shared library, using special configuration API provided by the Diameter Stack.

### 4.2.3. Interaction of Diameter Stack with Sh and Cx/Dx Modules

Cx/Dx and Sh Modules are clients of Diameter Stack. These modules make direct function calls into the Diameter Stack shared library. The Diameter Stack also calls one function from each of these modules. During their initialization, both Cx/Dx and Sh modules register a Message Processor function with the Diameter Stack. During registration, they provide their Application ID as well as the function pointer for the message processor callback function to the Diameter Stack. The Diameter Stack maintains an association between the Application ID and the Message Processor function associated with the application ID. Whenever a Diameter Message arrives from the network, which is to be processed locally and which is destined for an application ID for whom an Application has registered a message processor callback function – then the Diameter Stack fires the relevant callback function giving it the newly arrived Diameter Message, in the parameter. This way, the Diameter Stack also makes direct function calls into the Cx/Dx as well as the Sh Module.

## 4.3. HSS Main Module

The HSS Main module is an AMPS application. The main module is responsible for the following things:

- Loading the configuration file and initializing the application/server configuration structures as configured in the configuration file.
- Initializing and configuring the AMPS framework.
- Initializing and configuring the Diameter Base Protocol Stack.
- Initializing and loading each AMPS module required by HSS, i.e. Cx/Dx module, Sh Module, and the database module.
- Passing control over to AMPS framework for the event based logic to take over control and drive subsequent code execution.

The main module therefore makes direct function call into AMPS and Diameter Stack.

## 4.4. Cx/Dx Module and Sh Module

The Cx/Dx Module and Sh Module are implemented as AMPS modules. The Cx/Dx module implements 3GPP specified Cx/Dx Multimedia application while Sh module 3GPP defined Sh application implements Sh application interface component for HSS.

As part of their initialization, they register their Application IDs with the Diameter Stack as well as Message Processor callback function. Whenever a message for local processing arrives from the diameter peer that is destined for either of them, the Diameter stack calls the relevant callback function of the relevant module. From that point onwards, the control has been passed to the Cx/Dx or Sh module's message processor function. The message processor function then implements the required application logic to process the message that it has received.

### 4.5. Database Module

Database module is an AMPS module that is responsible for providing DBMS related services to Cx/Dx and Sh Modules. Whenever the Sh or Cx/Dx module require to interact with the DBMS, for getting or setting values in a certain table, they always do it through the Database module and never directly. In that sense, both the Sh and the Cx/Dx modules are clients of the Database module. The communication between the Sh or Cx/Dx module and the Database module is done through AMPS events. When the Sh or Cx/Dx module needs to query the DBMS or set values into the DBMS, they send an AMPS event to the Database module with all the required information. The Database module receives the event, and gets the associated event data, on the basis of which it interacts with the DBMS on behalf of the Sh or the Cx/Dx module. Similarly, the result of the requested Database operation is passed back to the requesting Sh or Cx/Dx module through AMPS events.

### 5. Typical control and Data Flow in HSS

A Typical control and data flow sequence is given below for better architectural understanding:

1. HSS Main module reads its configuration XML file. It then loads and initializes AMPS , Diameter Base protocol shared library and required AMPS modules, i.e. Cx/Dx module, Sh Module, and Database module. It transfers control to the AMPS scheduler.
2. As part of its initialization, each diameter Application registers a callback function with the Base protocol library against its Application ID.
3. A raw diameter message is received by Diameter base protocol library. The library converts it into structured format. Then the library passes this structured message to the relevant Diameter Application by calling its callback function. The callback function is found by doing a lookup against the Application ID in diameter message, in a library maintained mapping data structure.
4. The Diameter application's callback function does some processing based on message contents and application defined logic. As part of the application defined logic, a query needs to be made to get/set some data to/from the DBMS. Since this is a blocking operation, an event is generated. The associated parameters) are passed as event data. SessionID is also passed for correlation . The diameter application has already registered for the event it expects the DBMS to generate as a result of its own event. The processMessage callback function returns. The application callback also returns control.
5. The Database Module was registered for the type of event generated by the diameter application in step 4 above. Its event handler gets called. The event handler gets the parameters, queries the DBMS and gets back results. It formats the results in the desired standardized format. Then it generates an event and passes the result in addition to SessionID correlation info to the event as event parameter. The event handler returns control.
6. The Diameter Application had registered itself for the type of event generated by Database Module in step 5. Its event handler gets called. Application does some processing based on the result of the database operation. Then let us say the application needs to send a diameter message to the originator of the message it received. It builds a diameter message by calling different functions (some inside the application and some from the base protocol library). Then it calls the base protocol library's send message function to send the message. The event handler returns control.

Typical Control And Data Flow in HSS

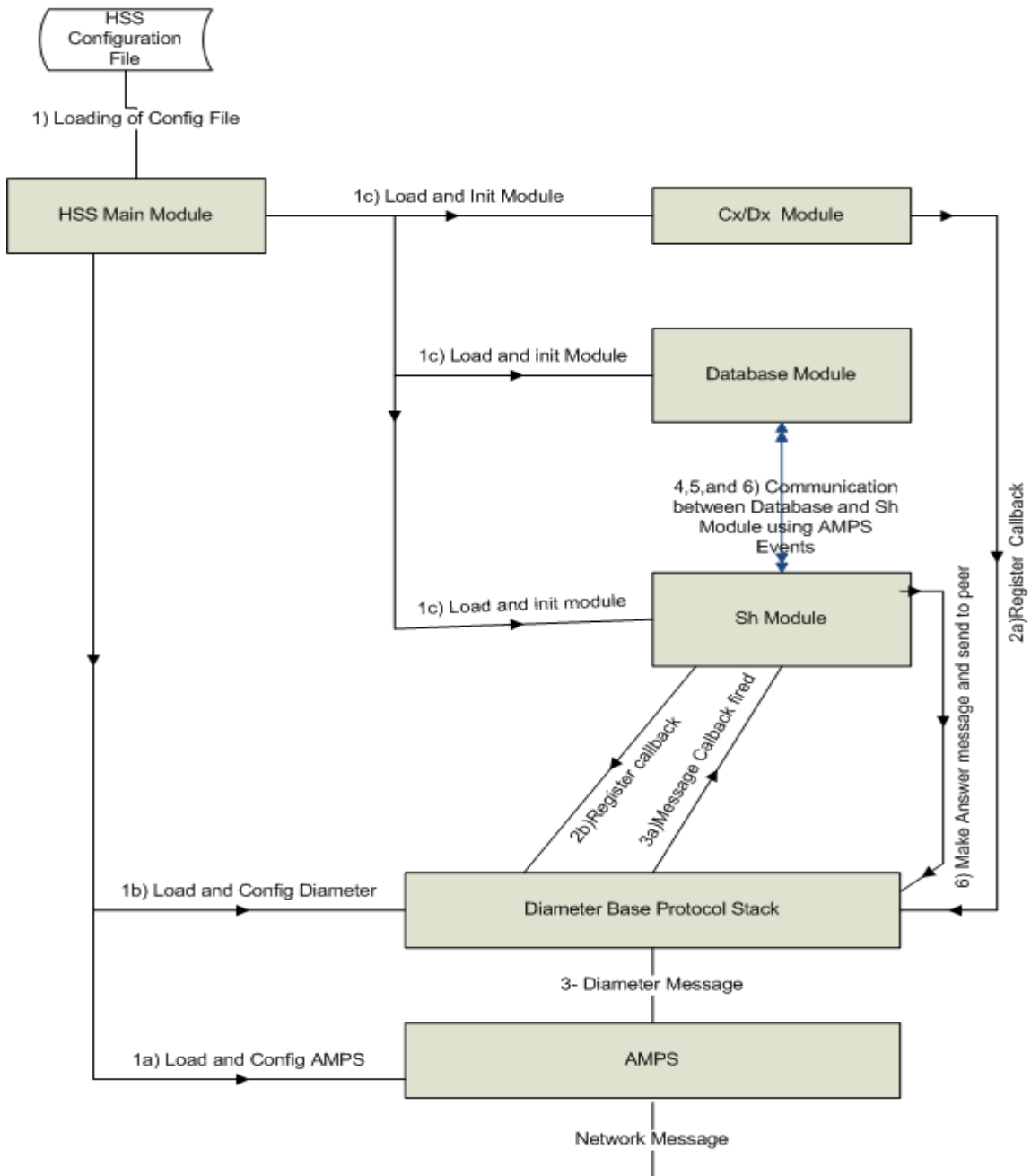


Fig: HSS Control Flow Diagram